Estudo Databricks

ESTUDO TEORIA

* Ferramenta de processamento de dados em nuvem de forma segura e otimizada
* Automatiza o ETL
* Plataforma de Lakehouse
* Dados de todos dos tipos (estruturados, não estruturados e estruturados)
* Bancos de dados relacionais são fonte de dados estruturados
* Abrange análise, engenharia e ciência de dados
* Databricks Community é a versão gratuita para os estudos
* SQL Analyse onde o usuário var ter acesso aos dados
* Compute é onde cuida do cluster
* Workflows a automação
* Cluster é um recurso computacional
* Single node, ou nó único de uma máquina
* Sequal = SQL
* CSV/JSON/TXT/SPORK} bronze > silver > gold > analise
* A qualidade dos dados é nesse sentido ->
* São as camadas de qualidade dos dados
* Traz confiabilidade para os dados
* BRONZE:
  + Replica os dados do arquivos no formato delta
* PRATA/SILVER:
  + 1:1 com a bronze, limpeza de estrutura
  + Nova nomenclatura
  + Filtros desejados
  + Também no formato delta
* GOLD:
  + Camada planejada pelo negócio
  + Olha para essa camada para fazer o BI e Data Science
  + Cruza informações e faz relacionamentos entre os a dados
  + Entrega o dado agregado
  + Camada que alimenta os times de negócio
* Entidades relacionais é a relação entre os dados
* Data Analyst Associate
  + Descrever o que é o Databricks SQL e seus benefícios
* Databricks sequencial é para ter melhor performance, simplificar a descoberta e o compartilhamento de novos insights, conecta com ISI e simplifica a governança e a administração
* Melhora em até 6x mais a performance e o preço do processamento dos dados
* Databricks sequencial é a melhor escolha pis a visualização é melhor, o preço o compartilhamento é automático para os alertas
* O Community é só para ver como funciona, não tem o catalog
* O cluster cai a cada 2h sendo assim preciso subir o arquivo novamente
* Metastore > Catalog > Schemas > Tabelas externas/Managed Tabelas/Views
* Metastore
  + Guarda dados ativos
  + Permissão
  + Criado com local de armazenamento padrão
* Catalog
  + Primeiro nível de organização
  + Usuários podem ver todos os catálogos onde USAGE é garantido
  + Pode ter vários Schemas com diferentes permissões
* Schemas
  + Segundo nível de organização
  + Pode ter tabelas externas, gerenciamento, views e cada um desses objetos pode ser configurado com diferentes permissões
* Managed Tables
  + Terceiro nivel de organização
  + Default comportamento de saber e criar conforme comandos
  + Delta é o formato suportado
* External Tables
  + Também é do terceiro nível
  + Guardada fora do gerenciamento
  + O Drop table não funciona aqui
  + Clona tranquilamente a tabela para um novo database sem modificar a original
  + Suporta: Delta, CSV, Parquet e etc
* Lakehouse Architectures
  + É um Data Sorce
  + Sistema em Nuvem
  + Bronze - ingestão dos dados
  + Silver – Limpeza dos dados e faz os joins
  + Gold – dado limpo, para a área de negócio
* Databricks SQL cluster é com photon
* Databricks SQL, usuários, benefícios, queries, dashboards, integração, parte conect, ingestão de dados e outras feramentas da BI. Lakehouse (arquitetura de medalha, streaming data)
* Usuários: descrever como a análise de dados e a engenharia interagem
* Benefícios: plataforma de processamento do Lakehouse e oportunidades para o BI
* Queries: descrever as queries como a peça central da escrita e execução do Databricks SQL code
* Dashboards: incluir várias queries em um dash e criar um do zero
* Compute: descrever o objetivo do warehouse, quando usar o serveless
* Partner Connet: conectar o databricks SQL com as ferramentas como firetram
* Ingestão de dados: como subir um arquivo pequeno, importar dados do objeto store
* Outras feramentas de BI: como conectar, identificar o databricks como complementar
* Medallion architecture: descrever, identificar qual camada é a mais comum para análise de dados
* Streaming data: descrever os benefícios e cuidados
* Managemen: delta lake, benefícios, tabelas, databses, views,data explorers, segurança (dono da tabela, PII data)
* Delta Lake básico: uma ferramenta para gerenciar arquivos e mapear meta dados
* Delta Lake benefícios: descrever os benefícios e capacidades
* Relacionais entidades: gerenciar e des-gerenciar as tabelas, locais de armazenamento das tabelas, criar e usar o drop tables, descrever a perspectiva de uma view e uma view temporária
* Data explorer: dar privilégios para um grupo, descrever os benefícios de gerenciar dados com o data explorer
* Table ownership: identificar o dono da tabela, mudar os acessos como também o dono de uma tabela
* PII data: identificar a adição de PII para uma organização específica
* SQL: básico (estruturação de query, combinação de dados e agregação), complex data (objetos dos dados, roll-ups, cubos)
* SQL in the Lakehouse (ANSI SQL, trabalhar com o nível server de dados, query históricas, usa as funções definidas e higher-order)
* Basic query structure: select, merge/insert, subqueries, messing data, joins, união/insert
* Agregations: simples e agregadas
* Nested data: acesso e explorar
* Window Functions: criar uma e usar
* Advanced grouping: roll-up
* ANSI – SQL: descrever e identificar os benefícios
* Silver level data: identificar e acessar
* Query history: descrever o propósito, identificar quando úteis
* High - order functions: aplicar e identificar do porquê usar
* User – defined functions: descrever o objeto e aplicar
* Visualização, capacidades, tipos, storytelling, dashboard (capacidades tipos, parametrização e queries de compartilhamento), produção (refresh, schedules e alertas)
* Capacidades: identificar quando usar o Dash, criar um schema básico e específico para visualização
* Customização: explicar como formatar e recuperar de novas visualizações, descrever como adicionar o visual através da formatação
* Storytelling: identificar a visualização pata comunicar de forma certa ideia
* Capacidades: criar um dash usando várias visualizações, utilizar um contador
* Parametrização: descrever como a query muda o output, identificar o comportamento dos parâmetros
* Sharing: descrever como compartilhar o dash, prós e contras de compartilhar o dash de diferentes formas
* Refresh schedules: descrever como configurar e como funciona
* Query alerts: como configurar um alerta básico, problemas com alertas
* Analytics applications: estatísticas descritivas e aplicações comuns como elt
* Estatística discreta: discreta X continua, descrever estatística discreta
* Summary statístics: mediana, moda e média
* Data enhacement: descrever como uma aplicação
* Data blending: descrever e misturar dados de fontes distintas
* ETL: fazer na gold
* Questões:
  + Para uma única query várias visualizações o dashboard é o ideal
  + O data explorer pode ser usado para ver uma tabela de meta dados, ver e mudar as permissões as tabelas

------------------------------------------------------------------------------------------------------------------------------

ESTUDO POR QUESTÕES

* A audiência chave do Databricks SQL é os analistas de dados e a audiência secundaria são os engenheiros de dados, cientistas de dados e profissionais de BI
* O Azure Databricks quando usado a tabela de visualização do para queries no SQL, permite reordenar manualmente, esconder e formatar as colunas de dados, porém não roda agregação de dados dentro do conjunto. Todas as agregações devem ser calculadas na consulta SQL antes de visualização
* Para renomear uma tabela no Databricks se utiliza ALTER TABLE RENAME TO
* No Unity Catalog do Databricks, o propósito de especificar um MANAGED LOCATION quando cria um catálogo ou schema é para definir o armazenado de objetos em nuvem para armazenar e gerenciar volumes, permitindo a organização e governança dos dados
* O aprimoramento de dados é aplicado em analises para enriquecer o conjunto de dados e obter insights mais profundos sobre os dados
* Otimizando a desempenho no Databricks que envolve operações complexas em matrizes utilizando Spark SQL, a função TRANFSORM seria a ideal para manipular da forma correta os arrays.
* O impacto em um painel se a taxa de atualização for definida como menos frequente do que o "Auto Stop" do SQL Warehouse, é que o dashboard irá continuar rodando no intervalo definido, e o SQL Warehouse também
* A diferença dos dados entre uma view temporária e uma view é que a view persiste após a seção mas ambas não guardam dados fisicamente
* Para configurar um agendamento de atualização é necessário clicar no Schedule na parte superior do dashboard e clicar em add schedule
* Os principais benefícios da implementar o delta lake na arquitetura é que ele providencia transações com ACID, metadados escaláveis e recursos de viagem no tempo permitindo acessar dados históricos
* MERGE INTO é adequado tanto para atualizar registros existentes quanto para inserir novos registros, dependendo se uma correspondência for encontrada na tabela de destino.
* INSERT INTO normalmente é usado para adicionar novos registros a uma tabela e não atualiza registros existentes.
* COPY INTO é um comando especializado usado no Databricks para carregar dados em uma tabela de fontes externas, como arquivos em um sistema de arquivos.
* ROLLUP é usado para agregação hierárquica de dados. Começa com o nível mais detalhado e avança para níveis mais amplos, terminando com um total geral (começa na coluna mais a esquerda do GROUP BY)
* COPY INTO é ideal para importar dados incrementais em massa no Databricks
* ANALYZE TABLE é usado para coletar estatísticas importantes sobre uma tabela. Seu foco principal é a coleta de métricas de dados para fins de otimização interna
* Com o dashboard-wide é possivel configurar todo o dashboard para que tenha as mesmas cores em toda a vizualização
* Grafico para mostrar o dados durante o tempo é o de linhas
* Os beneficios de trabalhar com dados de streaming é pode fazer em tempo real analises e tomar decisões e um cuidado é que existe altos indices de inconsistencia dos dados devido ao flow continuo
* As consultas SQL do Databricks fornecem um ambiente versátil dentro da plataforma Databricks para escrever, testar e executar código SQL  para a analise de dados e procesamento
* Ao gerir dados PII, é essencial que as organizações considerem o ambiente legal e regulamentar em que operam. Isto muitas vezes requer a adaptação de protocolos de tratamento de dados para cumprir com diversas leis
* O comando DESCRIBE HISTORY é utilizado para recuperar informações sobre cada operação de gravação em uma tabela Delta, mostrando a versão, carimbo de data/hora, usuário e outros detalhes da operação.
* Os passos essenciais para executar uma query básica no SQL seria abrir o SQL editor, selecionar um SQL Warehouse, especificar a query e rodar
* O propósito do Databricks Warehouse no ambiente de análise de dados é facilitar querys e analises, oferecendo gerenciamento para rodar queries em datasets grandes
* As responsabilidades primárias de um table owner no databricks é gerenciar os acessos dos usuários e permissões
* Para determinar se uma tabela é gerenciada ou não é possível ver pelo local em que se encontra os arquivos na definição da tabela
* Momentos chaves para a distribuição estatística média, variância, assimetria e curtose
* Quando temos dados de diferentes locais é crucial a unificação e sincronização deles para manter a consistência dos dados
* O primeiro passo para fazer uma apresentação com dados é fazer a query no SQL para garantir que tem o dado relevante
* Para fazer um COPY INTO com cabeçalho no final do código tem que ter o header = true
* A forma mais eficiente de limpar uma base que você terá que desduplicar é criando uma nova Window no nível silver
* A formatação correta dos dados no dashboard pode possibilitar a melhor compreensão dos dados
* Os dados discretos são quantitativos e podem assumir apenas valores fixos e distintos. Esses valores são contáveis, mas não necessariamente números inteiros.
* Os dados continuos são frequentemente visualizados usando gráficos de barras, gráficos de pizza ou gráficos de dispersão.
* Para intregrar o Databricks com ferramentas de BI (Power BI, Tableau, Looker)  o passo a seguir em comum ara conectar é  JDBC/ODBC (é um método de autenticação)entre o Databricks e a ferramenta
* O dashboard no databricks é acessível para varios usuários, sendo eles, analistas, gerentes e etc
* Para fazer um schema basico o primeiro passo é esquecrever a query do SQL
* O Dashboard do databricks é dinâmico e a lista suspensa funciona como um filtro
* Personally Identifiable Information (PII)
* O Pll no Databrciks geralmente é tratado atravez do Delta Lake para dar controles de acessos finos
* Cluster maiores oferecem um desenpenho melhor porém um custo mais elevado, e cluster menores apesar de econômicos tem desempenho pior.
* Query Based Dropdown List tem um funcionamento dinamico que gera uma lista de dropdwn baseada nos autputs distintos
* A forma correta de utilizar o window\_time é colocando (Coluna, 'Tempo desejado')
* Toda vez que tiver um AVG, COUNT, OVER, SUM tem que ter um GROUP BY
* As tabelas gerenciadas tem seus dados e metadados gerenciados pelo databricks, e armazenando em um local padrão, e a tabela não gerenciada tem somente seus metadados gerenciados, sendo assim também armazenados em um local padrão, e os dados são armazenados em um local especificados pelo usuário.
* A arquitetura de medalhão no databricks é uma abordagem para processar e organizar dados no lakehouse, usando uma camada bronze(dados brutos) > silver(dados limpos e coformados, o ETL)  > gold (tabelas de níveld e negócio e curadoria)
* As estatísticas descritivas são cruciais para fornecer uma visão geral rápida das tendências e características dos dados, como tendência central, dispersão e distribuição. Eles são o primeiro passo na análise de dados, proporcionando uma compreensão básica antes de prosseguir para uma análise inferencial mais complexa. Usadas pata somar e descrever o futuro principal do dataset
* No Databricks, a ferramenta Catalog Explorer permite aos usuários visualizar várias propriedades de tabelas, incluindo as informações de propriedade. Para acessar ela basta clicar na tabela
* No contexto de analise de dados um exemplo de aprimoramento eficaz dos dados seria adicionar dados meteológicos para ver a correlação do tempo com as vendas
* Os parâmetros de consulta são usadas para criar consultas SQL flexíveis e interativas. Eles atuam como espaços reservados na estrutura de consulta SQL, onde os valores reais podem ser fornecidos dinamicamente pelo usuário ou por outra fonte. Quando um usuário interage com o painel e altera um parâmetro de consulta, a consulta SQL se ajusta automaticamente para incluir esse novo valor.
* Para lidar com a importação de pequenos arquivos de texto, a abordagem recomendada para otimizar o desempenho é usar tecnicas desenvolvidas especialmente para pequenos arquivos.
* UI de upload de arquivo e UI de adição de dados: o Databricks oferece uma interface amigável para ingestão de dados, incluindo a UI de upload de arquivo e a UI de adição de dados. Esse recurso é particularmente útil para pequenos arquivos locais e oferece suporte a diversas necessidades de ingestão, incluindo ingestão de diferentes fontes de dados, como Azure Data Lake Storage, Amazon S3, Kafka e Kinesis.
* Trabalhando com arquivos de espaço de trabalho: Os arquivos de espaço de trabalho são especialmente recomendados para pequenos arquivos de dados, principalmente para fins de desenvolvimento e teste.
* Técnicas de otimização: Configurações como autoOptimize.optimizeWrite e autoOptimize.autoCompact, que podem otimizar a manipulação de arquivos pequenos durante os processos de ingestão e consulta. Essas configurações ajudam a melhorar o desempenho e a gerenciar dados com mais eficiência.
* Os dashboards no Databricks servem primeiramente para mostrar os resultados de multiplas queries simultaneamente em uma unica visão.
* O beneficio de utilizar a combinação de dados é que assim conseguimos ganhar inshgts
* Para ter a melhor aborgadem fazendo um dashboard ser minimalista e ter uma consistencia nas cores ajuda a focar nos dados.
* A vantagem primaria de se utilizar um serveless Databricks SQL é endpoint ou warehouse é que ele possibilita começos rapidos, redução significativa no tempo de fazer queries e analise de dados.
* Outras vantagens do serveless warehouse são: acesso instantâneo á computação (inicio rapido de consultas no SQL e cagras no BI), otimização de custos e capacidade, facil interação com ferramentas de BI, reinicialização automática, configuração segura e gerenciada (com trees camadas de isolamento).
* Os tipos de visualizaçoes que podemm ser feitos no Databricks SQL são grafico de linha, grafico de bolhas, contador e word cloud.
* A consulta usa LEFT JOIN, que retorna todos os registros da tabela esquerda (Funcionários) e os registros correspondentes da tabela direita (Departamentos). Se não houver correspondência, o resultado será NULL no lado da tabela direita. Portanto, todos os funcionários, incluindo John Smith (com departamento NULL), estarão na saída.
* CROSS JOIN, que combina cada linha de uma tabela com cada linha de outra tabela, frequentemente usada para criar combinações ou permutações.
* INNER JOIN, pega o que tem em comum nas duas.
* Os alertas e gatilhos são consigurados pelo usuário, podendo ser um variedade de canals
* CUBE produz todos os subtotais possíveis e o ROLLUP somente da subtottais de hierarquia
* O databricks pode receber varios arquivos ao mesmo tempo, desde que todos sejam do mesmo tipo.
* Para compartilhar um dashboard com colegas, basta compartilhar o link
* Quando o  databrciks esta conectado com outras ferramentas de BI, ele age como uma ferramenta de processamento de queries
* Widged é especifico para vizualizações individuais, aplicados somente á consulta dessa vizualização
* Parametros de dashboard são mais versateis e podem ser aplicados a diversas vizualizações
* Valores estaticos substituem a necessidade de paremetro e são usados para codificar um valor, quando um valor estático é usado, deixando de ser interativo e desaparecendo do painel ou widget
* Quando uma consulta é executada várias vezes, o Databricks armazena os resultados e os dados intermédios em cache. As execuções subsequentes da mesma consulta ou com cálculos semelhantes podem aproveitar esses dados armazenados em cache, levando a uma latência de consulta significativamente reduzida. Este ganho de eficiência é particularmente notável em consultas complexas em grandes conjuntos de dados, onde o acesso aos resultados armazenados em cache evita a necessidade de processamento e computação repetidos de dados.
* O acesso minimo que o usário tem que ter para consigurar um agendamento de atualização é de Edição
* A função primária do catalog explorer no databricksk é explorar dados e configurar acessos e controles
* Para modificar os acessos em uma tabela no databricks usando o catalog explorer ele seleciona a tabela, navega ate a aba de permissões e modifica
* O delta lake gerenciona os metaddaos no databricks mantendo o log de transações que guardam as mudanças nos metadados
* No Databricks Unity Catalog, que dá suporte a um namespace de três níveis (catalog.database.table), o comando SQL correto para criar uma nova tabela com base em uma consulta SELECT de uma tabela existente envolve a especificação do catálogo, do banco de dados e dos nomes da tabela.
* Um cenario em que os dados podem ser melhorados significativamente é em um uma campanha de marketing, enriquecer os dados com os dados demograficos e pyscograficos permitindo uma publicidade mais direcionada
* A sequência correta para executar uma consulta SQL no Databricks começa com a abertura do Editor SQL. Em seguida, você seleciona um warehouse SQL onde a consulta será executada. Depois disso, você constrói e edita sua consulta SQL diretamente no editor, que oferece suporte a recursos como preenchimento automático.
* Para setar a localização de uma tabela no databricks quando criando ou editando ela, se utiliza CREATE TABLE ... LOCATION e ALTER TABLE ... SET LOCATION
* No Databricks SQL, os painéis podem ser configurados para serem atualizados automaticamente em intervalos específicos. Isso é feito agendando o painel para atualizações regulares. Você pode configurar isso clicando em “Agendar” na parte superior da página do painel e selecionando “Adicionar agendamento”. Aqui você pode escolher o intervalo para atualização automática, como a cada hora. Além disso, você pode modificar o nome do agendamento e especificar um armazém SQL para executar as consultas do painel durante a atualização. Este recurso garante que o painel exiba informações atualizadas com base nos dados mais recentes
* Quando usamos o editor de consultas no Databricks é esperado encontrar databases disponiveis, tabelas, colunas e seus tipos de dados
* Para melhor descrever o ETL de ultimos detalhes podemos dizer que ele conduz o final da transformação dos dados para um projeto especifico, envolvendo ajustes finos
* Um gráfico de caixa é a escolha ideal para visualizar a distribuição de um conjunto de dados e é particularmente útil para identificar valores discrepantes, medianas e quartis. Neste cenário, mostraria efetivamente como as pontuações de feedback dos clientes estão distribuídas pelas diferentes categorias de produtos, destacando quaisquer pontuações incomuns (outliers) e dando uma imagem clara da tendência central e da variabilidade dentro de cada categoria.
* O Delta lake mantem, um histórico dos dados, mantendo o versionamento histórico e os dados, em um periodo configurado
* O benefício de usar ANSI SQL como linguagem de consulta padrão na arquitetura Lakehouse, como Databricks, reside em sua compatibilidade e interoperabilidade entre vários sistemas de banco de dados. ANSI SQL fornece uma sintaxe uniforme e um conjunto de recursos, garantindo que as consultas e operações SQL sejam consistentes e portáveis entre diferentes sistemas de banco de dados compatíveis com SQL. Essa padronização simplifica o gerenciamento de dados, o desenvolvimento de consultas e a integração com outros sistemas, aumentando a eficiência das operações de dados em um ambiente Lakehouse.
* O maior beneficiod e usar o Hadoop cluster é que ele tem um processamento mais rapido do que a ferramenta tradicional de Hadoop
* No cenario de uma manufatura que deseja ver os dados de produção o maior beneficio do databricks é que ele pode ser usado para gerar dados em tempo real
* Uma query para somar os saldos pela data do produto, categoria e mes, é utilizado o GROUP BY
* Para pegar dados de uma plataforma de midias socias, por exmplo, se deve primeiro usar o Partner Connect para pegar os dados e conectar com o databricks para procesamento
* Para otimizar uma query no Databricks uma opção é dividir os dados em menores partes
* O camada Gold é uma view dos dados que vem de diversas fontes
* Z-Ordering é uma técnica que reordena os dados em uma tabela Delta Lake com base nos valores de uma ou mais colunas. Isso é feito de forma que os dados com valores semelhantes nas colunas especificadas sejam armazenados fisicamente próximos uns dos outros. Como resultado, quando uma consulta é filtrada nas colunas especificadas, apenas os dados relevantes precisam ser lidos no disco, levando a melhorias significativas no desempenho da consulta.
* O Data Explorer recursos de intreface visual que facilitam no gertenciamento e organização de tabelas, permite os usuarios que criem e editem queries no SQL e também oferece ferramentas de vizualiação dos dados assim podem simplificar o gerenciamento de dados
* O Delta sterams provem os dados em tempo real, atualizando continuamente os dados
* O analista de dados pode usar ACLs para restringir o acesso aos dados PII, impondo a propriedade da tabela. As listas de controle de acesso permitem que o analista de dados configure permissões e gerencie o acesso à tabela com base na identidade do usuário, associação ao grupo ou endereço IP. As ACLs fornecem uma maneira flexível de impor políticas de segurança no nível da tabela ou do objeto, e o analista de dados pode usá-las para impedir o acesso não autorizado aos dados PII.
* ALTER COLUMN é usada para modificar o tipo de dados de uma coluna existente em uma tabela.
* O formato de arquivo Parquet é um formato de armazenamento colunar que fornece técnicas eficientes de compactação e codificação, o que o torna ideal para armazenar e consultar grandes conjuntos de dados. Além disso, o particionamento por hash fornece uma distribuição uniforme de dados entre partições com base no valor de hash da coluna de particionamento. Este método garante que os dados sejam distribuídos uniformemente em todas as partições, o que, por sua vez, ajuda a reduzir o tempo de consulta e a melhorar o desempenho geral da tabela com filtragem e agregação eficientes.
* Os usuários podem realizar computação distribuída em conjuntos de dados consideráveis que não cabem na memória usando o recurso de computação do Databricks SQL. O analista de dados pode processar com eficiência o grande conjunto de dados e realizar análises SQL do Databricks usando computação distribuída.
* A camada Silver é onde os dados são transformados e processados para uso analítico
* Delta Lake é uma camada de armazenamento transacional que fornece conformidade ACID para processamento de dados em Databricks SQL.
* Para processamento de big data, a camada de armazenamento conhecida como Delta Lake oferece transações ACID e evolução de esquema. A evolução do esquema refere-se à capacidade de modificar o esquema de uma tabela sem a necessidade de reescrevê-la completamente. Isto pode ser útil se o esquema da tabela precisar ser atualizado para incluir novas fontes de dados ou corrigir erros no esquema atual. É significativo observar que apenas a evolução do esquema compatível com versões anteriores é suportada pelo Delta Lake. Isso significa que os dados gravados usando o esquema antigo devem ser legíveis pelo novo esquema. Delta Lake não dá suporte a alterações de esquema, como alterar a ordem das colunas ou alterar o tipo de dados de uma coluna de uma forma que impossibilite a leitura dos dados existentes.
* A camada de dados mais utilizada pelos analistas de dados é a Gold
* Local onde se pode escreverr e executar queries é no SQL Editor
* O Databricks é uam ferramenta complementar para as ferramnestas de BI, deixando o processo mais rapido
* Para conectar o Databricks com o Fivetran, se pode usar o Partner Connects ára automatizar um workflow que estabiliza um cluster para o Fivetran interagir
* Os engenheiros de dados são os principais responsáveis por construir, gerenciar e otimizar pipelines e arquiteturas de dados. Eles usam o serviço Databricks Data Science and Engineering para executar tarefas como ingestão de dados, transformação, qualidade e governança. Os engenheiros de dados podem usar o Databricks SQL como um serviço secundário para consultar, analisar e visualizar dados do lakehouse, mas este não é o seu foco principal.
* Para diminuir o tempo de processsamento de queries uma estratégia usada pode ser usar o serveless SQL
* Quando temos dados que atualizam a cada minuto e usuário deseja que o dashboard atualize na mesma frequência temos que verificar qual ser ao custo desse recurso computacional
* Tabelas externas são tabelas definidas no metastore do Databricks usando as informações armazenadas em um local de armazenamento de objetos em nuvem. As tabelas externas não gerenciam os dados, mas fornecem um esquema e um nome de tabela para consultar os dados. Para criar uma tabela externa, você pode usar a instrução CREATE EXTERNAL TABLE e especificar o caminho de armazenamento do objeto para a cláusula LOCATION.
* As caixas de texto baseadas em Markdown são úteis como rótulos em um painel. Eles permitem que o analista de dados adicione texto a um painel usando o comando mágico %md em uma célula do notebook e, em seguida, selecione o ícone do painel no menu de ações da célula.
* O Databricks SQL tem espaço para queries, vizualizações e pode ser feita serveless
* Quando se deleta uma tabela que é externa o seu metadado não é apagado
* Os analistas que mexem com PLL devem considerar as melhores praticas da organização referente a segurança, questões legais e onde a analise esta sendo feita.
* Delta Lake armazena dados de tabela como uma série de arquivos de dados em um local especificado, mas também armazena metadados de tabela em um log de transações. Os metadados da tabela incluem o esquema, informações de particionamento, propriedades da tabela e outros detalhes de configuração.
* Um data lakehouse baseado em Delta Lake é uma arquitetura de plataforma de dados que combina a escalabilidade e a flexibilidade de um data lake com a confiabilidade e o desempenho de um data warehouse. Uma das principais vantagens de usar um data lakehouse baseado em Delta Lake em vez de soluções comuns de data lake é que ele oferece suporte a transações ACID, que garantem a integridade e consistência dos dados.